Toward A Good Algorithm for Determining Unsatisfiability of Propositional Formulas

John Franco* R. Swaminathan†

May 20, 1997

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

19970609 025

Abstract

We present progress toward an algorithm that provides short certificates of unsatisfiability with high probability when inputs are random instances of 3-SAT. Such an algorithm would incorporate an approximation algorithm $\mathcal{A}$ for the 3-Hitting Set problem. Using $\mathcal{A}$ it would determine an approximation for the minimum fraction of variables that must be set to *true* (*false*) in order to satisfy the positive (negative) clauses. If the fraction is high enough, then the instance is deemed unsatisfiable.

Key words Satisfiability, Resolution, Theorem Proving, Hitting Set

1 Introduction

It is well known that the problem of determining the existence of a satisfying truth assignment for a given propositional formula in Conjunctive Normal Form (CNF) is NP-complete. If clauses have exactly three literals each, the problem is called 3-SAT and this problem is also NP-complete. However, there exist polynomial time algorithms that, under certain circumstances, can produce a solution to a random satisfiable instance of 3-SAT with high probability. This paper is concerned with the question of the existence of a polynomial time algorithm that, with high probability, verifies the unsatisfiability of a random unsatisfiable instance of 3-SAT.

Let $\mathcal{I}$ be a random CNF Boolean expression where each of m clauses has exactly 3 literals taken uniformly and independently from a set V of n Boolean variables and complemented independently with probability 1/2. Below, we refer to this model of generation of a random instance as $M(m, n, 3)$. Suppose m/n is held constant as m and n tend to ∞. A series of papers [7, 3, 5, 10] ended with the currently best result ([10]) that $\mathcal{I}$ is unsatisfiable, with probability tending to 1, if $m/n > 4.75$. Another series of papers [1, 2, 3, 8] ended with the currently best result ([8]) that $\mathcal{I}$ is satisfiable,

*Computer Science Department, University of Cincinnati, Cincinnati, Ohio 45221 (franco@franco.csm.uc.edu).
Supported in part by the Office of Naval Research: N00014-94-1-0382
†Computer Science Department, University of Cincinnati, Cincinnati, Ohio 45221 (swamy@jupiter.csm.uc.edu).

DTIC QUALITY INSPECTED 3

and a satisfying solution to $\mathcal{I}$ may be found in polynomial time, with probability tending to 1, if $m/n < 3.003$. Thus, random satisfiable instances of 3-SAT are usually easily solved for all but a small range of values of the ratio m/n generating mostly satisfiable instances. On the other hand, there is no known polynomial time algorithm that almost always verifies unsatisfiablity when m/n is a constant greater than 4.75. Moreover, in [4] it is shown that resolution (therefore, other well known methods for solving Satisfiability including the Davis-Putnam procedure) requires exponential time, almost always, to verify unsatisfiability for all constant $m/n > 4.75$.

A positive result for verifying unsatisfiability, if one exists, is clearly much tougher to find than the positive results for determining satisfiability cited above. A reasonable candidate algorithm probably should avoid a search over many truth assignments to determine that none will satisfy a given instance. This paper presents a reasonable strategy. The idea is to recast 3-SAT as a 3-Hitting Set problem and use an approximation algorithm for the 3-Hitting Set problem to prove unsatisfiability. An instance of the 3-Hitting Set problem is a set S of atoms and a collection of triples $\mathcal{T} = \{T : T \subset S, |T| = 3\}$. The problem is to find the minimum $S' \subset S$ such that for every $T \in \mathcal{T}$, there is an $s \in T$ which is also in S'. Any subset $S'' \subset S$ that satisfies the above condition is called a hitting set, and S' is an optimal hitting set. We present the mechanics of the method, demonstrate its feasibility, and show how close we have come to its realization.

2 Unsatisfiability as a 3-Hitting Set problem

The idea is as follows. Given a random instance $\mathcal{I}$ of 3-SAT, keep only the positive and negative clauses (those that have all literals positive or all literals negative). Determine the minimum number of variables that must be set to *true* to satisfy the positive clauses. Determine the minimum number of variables that must be set to *false* to satisfy the negative clauses. If the sum of the two numbers is greater than n, then at least one variable must be set to *true* and *false* if all the positive and negative clauses are to be satisfied. Since this is impossible, $\mathcal{I}$ must be unsatisfiable if the sum is greater than n.

The problem of determining the minimum number of variables that must be set to *true* or *false* is equivalent to a 3-Hitting Set problem where the given set of atoms is the set of variables and the sets composed from atoms are the clauses. Unfortunately, the 3-Hitting Set problem is NP-complete. However, if there is an approximation algorithm for 3-Hitting Set with a certain performance guarantee, then it is possible to decide unsatisfiability anyway. The sections below discuss the liklihood that such an approximation algorithm exists and show, if one does exist, how to use it assuming inputs are generated according to $M(m, n, 3)$.

3 Properties of random 3-SAT instances

This section develops the probabilistic analysis of random instances of 3-SAT generated according to $M(n, m, 3)$ and shows how a polynomial time approximation algorithm for an optimization problem known as 3-Hitting Set can be used to verify unsatisfiability with probability tending to 1.

Lemma 1: With probability tending to 1, for any $\epsilon > 0$, an instance $\mathcal{I}$ of 3-SAT generated according to $M(n, m, 3)$ has at least $(m/8)(1-\epsilon)$ negative clauses and at least $(m/8)(1-\epsilon)$ positive clauses.

Proof: It is sufficient to prove the hypothesis for positive clauses only since the result for negative clauses is identical and, if the probability of two events tends to 1, then the probability of the intersection of those two events also tends to 1. The probability that $\mathcal{I}$ has at least r positive clauses is given by

$$Pr(\mathcal{I} \text{ has } \geq r \text{ positive clauses}) = \sum_{k=r}^{m} \binom{m}{k} (1/8)^k (7/8)^{m-k}.$$

This is a binomial distribution with mean $m/8$. Setting $r = (m/8)(1-\epsilon)$ and using the well-known Chernoff bound on the lower tail of a binomial distribution we can bound the sum from below as follows:

$$Pr(\mathcal{I} \text{ has } \geq (m/8)(1-\epsilon) \text{ positive clauses}) \geq 1 - e^{-(\epsilon)^2 (m/8)/2}.$$

This tends to 1 with increasing m and the lemma is proved. □

Lemma 2: With probability tending to 1, the minimum fraction α of variables that must be set to *true* (*false*) to satisfy all the positive (negative) clauses of $\mathcal{I}$ is at least the value given by

$$\beta = m/n = \frac{8}{1-\epsilon} \frac{\alpha \ln(\alpha) + (1-\alpha)\ln(1-\alpha)}{\ln(1-(1-\alpha)^3)}.$$

Proof: Consider only the positive clauses as the case of negative clauses is identical. Let $V'(\alpha) = \{v_1, v_2, ..., v_{\lfloor \alpha n \rfloor}\}$ be a random subset of $\lfloor \alpha n \rfloor$ variables taken from V. The probablility that setting only variables in $V'(\alpha)$ to *true* satisfies r positive clauses is

$$\left(1 - \frac{\binom{n-\lfloor \alpha n \rfloor}{3}}{\binom{n}{3}}\right)^r.$$

The average number of sets $V'(\alpha)$ that satisfy the positive clauses is

$$\binom{n}{\lfloor \alpha n \rfloor} \left(1 - \frac{\binom{n-\lfloor \alpha n \rfloor}{3}}{\binom{n}{3}}\right)^r.$$

This is an upper bound on the probability that there exists a set $V'(\alpha)$ that satisfies the positive clauses. We need to find the maximum α for which this expression tends to 0. Simplifying by using Stirling's approximation for factorials, and substituting $(m/8)(1-\epsilon)$ for r since, from Lemma 1, we have at least that many positive clauses, we need to find the maximum α such that

$$\frac{(1-(1-\alpha)^3)^{(1-\epsilon)(m/8)}}{\alpha^{\alpha n}(1-\alpha)^{(1-\alpha)n}} = \left(\frac{(1-(1-\alpha)^3)^{(1-\epsilon)(m/n)/8}}{\alpha^{\alpha}(1-\alpha)^{(1-\alpha)}}\right)^n \to 0.$$

This is satisfied if

$$\beta = m/n > \frac{8}{1-\epsilon} \frac{\alpha \ln(\alpha) + (1-\alpha)\ln(1-\alpha)}{\ln(1-(1-\alpha)^3)}.$$

The lemma follows. □

A result similar to Lemma 2 is proved in [6].

From Lemma 2, with probability tending to 1, if $\beta > 41.52$, then the number of variables that must be set to *true* to satisfy the positive clauses and the number of variables that must be set to *false* to satisfy the negative clauses both must be greater than $n/2$. Consequently, with probability tending to 1 and $\beta > 41.52$, a random instance of 3-SAT is not satisfiable and that fact may be verified in polynomial time if there is a fast algorithm for finding the minimum number of *true* (*false*) variables needed to satisfy the positive (negative) clauses.

The problem of finding the minimum number of *true* variables needed to satisfy the positive clauses is equivalent to the problem of finding a minimum 3-Hitting Set for a collection of triples that is in one-one correspondence with the clauses. Although this problem is NP-complete, if there is a good enough polynomial time approximation algorithm for 3-Hitting Set, we can use it to reliably verify unsatisfiability in polynomial time for large, but constant ratios β. By reliably verify unsatisfiability in polynomial time we mean the algorithm provides a polynomial time test of unsatisfiability which, if successful, proves a given instance is unsatisfiable and is not successful with probability tending to 0. The question of precisely how good such an approximation algorithm must be to reliably verify unsatisfiability in polynomial time is answered after the following discussion.

Suppose there is a polynomial time approximation algorithm $\mathcal{A}$, for a 3-Hitting Set instance $\mathcal{H}$ with m triples taken from n atoms, that has the following approximation property: if the minimum hitting set for $\mathcal{H}$ is less than $n/2$, then $\mathcal{A}$ returns a hitting set of no more than $n\gamma_{\mathcal{A}}(m,n)$ elements. We can apply $\mathcal{A}$ to $\mathcal{H}$ and, if β is big enough to make $\alpha n > n\gamma_{\mathcal{A}}(m,n)$, then, with probability tending to 1, the minimum hitting set for $\mathcal{H}$ is greater than $n\gamma_{\mathcal{A}}(m,n)$. Since $\mathcal{A}$ is an approximation algorithm, it returns a hitting set for $\mathcal{H}$ of size greater than $n\gamma_{\mathcal{A}}(m,n)$, with probability tending to 1. Due to the approximation property of $\mathcal{A}$, a returned hitting set of size greater than $n\gamma_{\mathcal{A}}(m,n)$ is not possible if the minimum hitting set for $\mathcal{H}$ is of size less than or equal to $n/2$. Hence, with probability tending to 1, for large enough β, $\mathcal{A}$ can be used to determine whether a set of positive clauses taken from a random instance of 3-SAT requires more than $n/2$ *true* variables to be satisfied. It follows that $\mathcal{A}$ can be used to reliably verify unsatisfiability in polynomial time for large enough β. It remains to determine what $\gamma_{\mathcal{A}}$ needs to be in order to support the above observation for constant β under model $M(m,n,3)$.

Theorem 3: Let $\mathcal{I}$ be an instance of 3-SAT generated from $M(m,n,3)$ and let β be the limiting ratio of m/n and suppose $\beta > 41.52$. Let $\mathcal{A}$ be a polynomial time approximation algorithm for 3-Hitting Set with $\gamma_{\mathcal{A}}(m,n)$ performance guarantee. Suppose there exists a function $c(\beta)$, $1/2 \leq c(\beta) \leq 1$, and $c(\beta)$ decreases with increasing β, such that, for very small $\epsilon > 0$,

$$\gamma_{\mathcal{A}}(m,n) < 1 - \sqrt{\frac{c(\beta)\ln(\beta(1-\epsilon)/8)}{\beta(1-\epsilon)/8}}.$$

Then, with probability tending to 1, $\mathcal{A}$ verifies that $\mathcal{I}$ is unsatisfiable.

Proof: We need to show that the right side of the equation of Lemma 2 is less than β after substituting $\gamma_{\mathcal{A}}$ for α. First, we do this with $c(\beta) = 1$ to provide an upper bound good for all β. We assume $\gamma_{\mathcal{A}} > 1/2$ since otherwise the theorem follows immediately. In what follows we ignore

ϵ, which is assumed to be a constant very close to 0, for simplicity.

$$\begin{aligned}
\beta &> 8\frac{\ln(\beta/8)}{(1-\gamma_{\mathcal{A}})^2} \quad \text{(by hypothesis)} \\
&> 8\frac{\ln(\frac{\ln(\beta/8)}{(1-\gamma_{\mathcal{A}})^2})}{(1-\gamma_{\mathcal{A}})^2} \quad \text{(by substitution)} \\
&= 8\frac{\ln(\ln(\beta/8)) - 2\ln(1-\gamma_{\mathcal{A}})}{(1-\gamma_{\mathcal{A}})^2} \quad \text{(by simplification)} \\
&> 8\frac{-\gamma_{\mathcal{A}}\ln(\gamma_{\mathcal{A}})/(1-\gamma_{\mathcal{A}}) - \ln(1-\gamma_{\mathcal{A}})}{(1-\gamma_{\mathcal{A}})^2} \quad \text{(since } (1-x)\ln(1-x) < x\ln(x) \text{ if } x > 1/2\text{)} \\
&= 8\frac{-\gamma_{\mathcal{A}}\ln(\gamma_{\mathcal{A}}) - (1-\gamma_{\mathcal{A}})\ln(1-\gamma_{\mathcal{A}})}{(1-\gamma_{\mathcal{A}})^3} \quad \text{(multiply top and bottom by } 1-\gamma_{\mathcal{A}}\text{)} \\
&> 8\frac{\gamma_{\mathcal{A}}\ln(\gamma_{\mathcal{A}}) + (1-\gamma_{\mathcal{A}})\ln(1-\gamma_{\mathcal{A}})}{\ln(1-(1-\gamma_{\mathcal{A}})^3)} \quad \text{(since } \ln(1-x) < -x\text{)}.
\end{aligned}$$

Since the last expression is the right side of the equation of Lemma 2 with $\gamma_{\mathcal{A}}$ substituted for α, we have $\gamma_{\mathcal{A}} < \alpha$.

Next, we show that $c(\beta) = 1/2$ is sufficient when β is large.

$$\begin{aligned}
\beta &> 8\frac{(1/2)\ln(\beta/8)}{(1-\gamma_{\mathcal{A}})^2} \quad \text{(by hypothesis)} \\
&> 8\frac{(1/2)\ln((1/2)\frac{\ln(\beta/8)}{(1-\gamma_{\mathcal{A}})^2})}{(1-\gamma_{\mathcal{A}})^2} \quad \text{(by substitution)} \\
&= 8\frac{(1/2)\ln((1/2)\ln(\beta/8)) - \ln(1-\gamma_{\mathcal{A}})}{(1-\gamma_{\mathcal{A}})^2} \quad \text{(by simplification)} \\
&> 8\frac{-\gamma_{\mathcal{A}}\ln(\gamma_{\mathcal{A}})/(1-\gamma_{\mathcal{A}}) - \ln(1-\gamma_{\mathcal{A}})}{(1-\gamma_{\mathcal{A}})^2} \quad \text{(for } \beta \to \infty, (1/2)\ln\ln(\sqrt{\beta/8}) > 1 > -\gamma_{\mathcal{A}}\ln(\gamma_{\mathcal{A}})/(1-\gamma_{\mathcal{A}})\text{)} \\
&= 8\frac{-\gamma_{\mathcal{A}}\ln(\gamma_{\mathcal{A}}) - (1-\gamma_{\mathcal{A}})\ln(1-\gamma_{\mathcal{A}})}{(1-\gamma_{\mathcal{A}})^3} \quad \text{(multiply top and bottom by } 1-\gamma_{\mathcal{A}}\text{)} \\
&> 8\frac{\gamma_{\mathcal{A}}\ln(\gamma_{\mathcal{A}}) + (1-\gamma_{\mathcal{A}})\ln(1-\gamma_{\mathcal{A}})}{\ln(1-(1-\gamma_{\mathcal{A}})^3)} \quad \text{(since } \ln(1-x) < -x\text{)}.
\end{aligned}$$

□

4 An approximation algorithm for 3-Hitting Set

In this section we present an approximation algorithm $\mathcal{A}$ for 3-Hitting Set with $\gamma_A(m,n) < 1 - 2\sqrt{3}/(9\sqrt{\beta})$ where m is the number of triples and n is the number of atoms from which triples

$\mathcal{A}(\mathcal{I})$:
Input: *Instance $(\mathcal{G}, A)$ of 3-Hitting Set: $\mathcal{G}$ is a collection of sets taken from the atom set A;*
Output: *A hitting set for $(\mathcal{G}, A)$;*

1. Set $\bar{S} = 3|\mathcal{G}|/|A|$.
2. Set $T = \emptyset$.
3. Repeat the following as long as $\bar{S} > 1$.
   (a) Choose an atom $a \in A$ such that $S(a) > 1$.
   (b) Set $T = T \cup \{a\}$.
   (c) Set $A = A - \{a\}$.
   (d) Set $\mathcal{G} = \mathcal{G} - \{B : B \in \mathcal{G} \text{ and } a \in B\}$.
   (e) Set $\bar{S} = 3|\mathcal{G}|/|A|$
4. Repeat the following as long as $\mathcal{G} = \emptyset$.
   (a) Choose a set $B \in \mathcal{G}$.
   (b) Choose an atom $a \in B$.
   (c) Set $T = T \cup \{a\}$.
   (d) Set $\mathcal{G} = \mathcal{G} - \{B : B \in \mathcal{G} \text{ and } a \in B\}$.
5. Output T.

Figure 1: Algorithm for finding 3-covers of 3-cover graphs

are taken. The algorithm, presented in Figure 1, is related to but weaker than the obvious greedy method as it only selects atoms that occur an average number of times among remaining sets instead of the maximum number of times. Although $\gamma_{\mathcal{A}}(m, n)$ is not enough to satisfy Theorem 3, we note that it is fairly close to what is needed. It is possible that a similar algorithm with a more accurate analysis will yield the required approximation result.

An unusual operation performed within $\mathcal{A}$ is computing the average number of sets containing a particular atom. Let A denote a set of atoms and $\mathcal{G}$ denote a collection of 3-subsets of A, and $(\mathcal{G}, A)$ denote an instance of 3-Hitting Set. Let $S(a)$ denote the number of sets in $\mathcal{G}$ containing atom a. The average number of sets containing a particular atom is $\bar{S} = \sum_{a \in A} S(a)/|A|$. At the outset, $\bar{S} = 3m/n$. Upon every iteration of the main loop of $\mathcal{A}$ the sets containing one of the more frequently occurring atoms are eliminated. This lowers $\bar{S}$ somewhat. However, computing $\bar{S}$ does not change from one iteration to the next: take the product of 3 and the ratio of number of sets to number of atoms not yet considered.

Theorem 4: Algorithm $\mathcal{A}$ always returns a hitting set for $(\mathcal{G}, A)$ and runs in time bounded by a polynomial in $|\mathcal{G}|$ and $|A|$.

Proof: We show the following loop invariant holds prior to each step of either loop: T is a hitting set for all sets eliminated from the original $\mathcal{G}$ up to the present. Clearly, this is true for the first iteration of the first loop. On every succeeding iteration, every eliminated set has at least one of its atoms placed in T. Correctness follows.

The total number of iterations is at most $|A|$. Each iteration takes $O(|\mathcal{G}|)$ time. Hence, $\mathcal{A}$ runs in polynomial time. □

Theorem 5: $\gamma_{\mathcal{A}}(m,n) < 1 - 2\sqrt{3}/(9\sqrt{\beta})$.

Proof: At iteration k of Step 3, the number of atoms remaining is $n-k$ and the number of sets remaining is no greater than $m\prod_{i=0}^{k-1}(1-3/(n-i))$. Hence, at iteration k of Step 3,

$$\bar{S} < \frac{3}{n-k} m \prod_{i=0}^{k-1} \left(1 - \frac{3}{n-i}\right).$$

We find the value of k that makes the right side of the inequality 1. This is an upper bound on the number of iterations taken by Step 3. It is sufficient to find k such that

$$\ln(3) + \ln(m) - \ln(n-k) + \sum_{i=0}^{k-1} \ln\left(1 - \frac{3}{n-i}\right) = 0.$$

Using $\ln(1-x) = -x - \Theta(x^2)$ if $|x| < 1$, the above can be written

$$\ln(3) + \ln(m) - \ln(n-k) - \sum_{i=0}^{k-1} \left(\frac{3}{n-i} + \Theta\left(\frac{1}{(n-i)^2}\right)\right) = 0.$$

Using $\sum_{i=0}^{k-1}(1/(n-i)) = \ln(n) + \tau + \Theta(1/n)$ and $\sum_{i=0}^{k-1}(1/(n-i)^2) = \Theta(1/(n-k))$, we have

$$\begin{aligned}
\ln(3) + \ln(m) - \ln(n-k) - 3\ln(n) + 3\ln(n-k) + \Theta(1/(n-k)) &= 0 \\
\ln(3) + \ln(m/n) - 2(\ln(n) - \ln(n-k)) + \Theta(1/(n-k)) &= 0 \\
\ln(3) + \ln(\beta) - \ln((n/(n-k))^2) + \Theta(1/(n-k)) &= 0 \\
\ln((n/(n-k))^2/3) + \ln(e^{\Theta(1/(n-k))}) &= \ln(\beta) \\
(n/(n-k))^2/3 &= \beta e^{-\Theta(1/(n-k))} \\
k &= n - n/\sqrt{3\beta} \quad \text{(for large } n\text{)}.
\end{aligned}$$

The size of $\mathcal{G}$ just before beginning Step 4 is, following steps similar to those above, no greater than

$$m \prod_{i=0}^{k-1} \left(1 - \frac{3}{n-i}\right) = \frac{m}{(3\beta)^{3/2}},$$

where $k = n\left(1 - \frac{1}{\sqrt{3\beta}}\right)$. Therefore, the total number of atoms used in the hitting set is no greater than

$$k + m/(3\beta)^{3/2} = n - \frac{n}{\sqrt{3\beta}} + \frac{\beta n}{(3\beta)^{3/2}}$$

$$= n - \frac{n}{\sqrt{3\beta}} + \frac{n}{3^{3/2}\sqrt{\beta}}$$
$$= n\left(1 - \frac{2\sqrt{3}}{9\sqrt{\beta}}\right) = n\left(1 - \frac{1}{\sqrt{\frac{27}{4}\beta}}\right).$$

Hence, $\gamma_{\mathcal{A}}(m,n) < 1 - \frac{1}{\sqrt{\frac{27}{4}\beta}}$. □

5 Is a better approximation algorithm possible?

Given the negative results on approximation algorithms for Hitting Set and other hard optimization problems found in [9], the question arises whether an approximation algorithm for Hitting Set satisfying the requirements of Theorem 3 is possible. In [9] it is shown that an approximation algorithm for Hitting Set with approximation ratio less than $c\log(n)$, for some c close to 1, is unlikely. However, 3-Hitting Set is MAX SNP-hard which means it can be approximated with constant factor in polynomial time. Moreover, it is felt that the guaranteed constant factor should be close to 1 ([11]). Hence, the existence of such an approximation algorithm is likely.

6 Acknowledgment

The authors would like to thank Ewald Speckenmeyer with suggesting algorithm $\mathcal{A}$ and its analysis.

References

[1] Chao, M. T., and Franco, J., "Probabilistic analysis of two heuristics for the 3-satisfiability problem," *SIAM J. Comput.* **15** (1986), 1106–1118.

[2] Chao, M. T., and Franco, J., "Probabilistic analysis of a generalization of the unit-clause literal selection heuristic for the k satisfiability problem," *Information Sciences* **51** (1990), 289–314.

[3] Chvatal, V., and Reed, B., "Mick gets some (the odds are on his side)," *Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science* (1992), 620–627.

[4] Chvatal, V., and Szemeredi, E., "Many hard examples for resolution," *J. ACM* **35** (1988), 759–768.

[5] El Maftouhi, A., and Fernandez de la Vega, W., "On random 3-SAT," (to appear).

[6] Fernandez de la Vega, W., Paschos, V. Th., and Saad, R., "Average case analysis of a greedy algorithm for the minimum hitting set problem," *Proceedings of LATIN* (1992), 130–138.

[7] Franco, J., and Paull, M., "Probabilistic analysis of the Davis-Putnam procedure for solving the satisfiability problem," *Discrete Applied Math.* **5** (1893), 77–87.

[8] Frieze, A., and Suen, S., "Analysis of simple heuristics for random instances of 3-SAT," (1992).

[9] Lund, C., and Yannakakis, M., "On the hardness of approximating minimization problems," *J. ACM* **41** (1994), 960–.

[10] Kamath, A., Motwani, R., Palem, K., and Spirakis, P., "Tail bounds for occupancy and the satisfiability threshold conjecture," *Stanford University Tech Report* (1994).

[11] Yannakakis, M., personal communication.